SYSTEM STACK AND HEAP

GIRISH S KUMAR

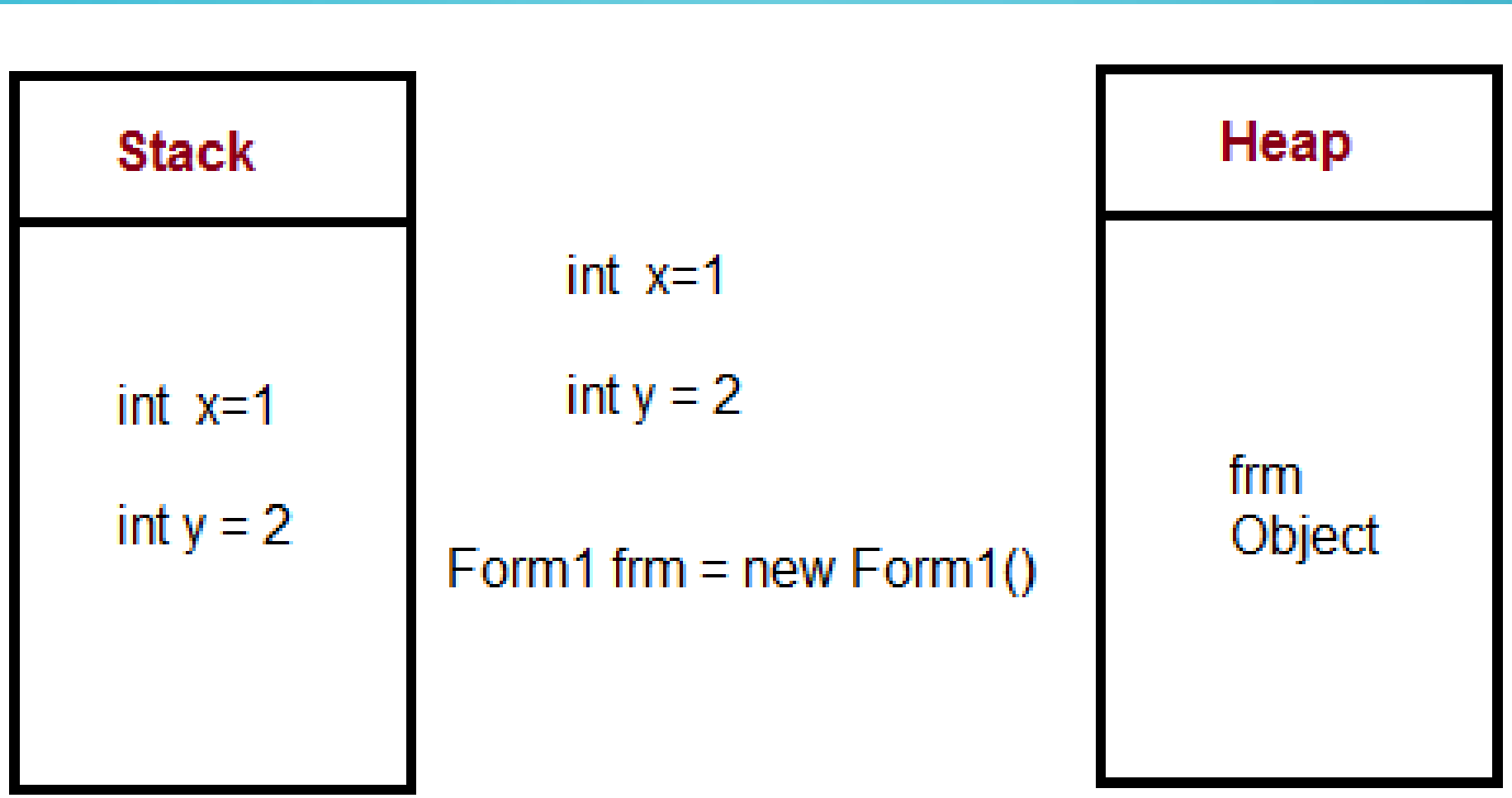
WHAT IS SYSTEM STACK AND HEAP

Stack is used for static memory allocation and Heap for dynamic memory allocation, both stored in the computer's RAM .

Variables allocated on the stack are stored directly to the memory and access to this memory is very fast, and it's allocation is dealt with when the program is compiled.

When a function or a method calls another function which in turns calls another function etc., the execution of all those functions remains suspended until the very last function returns its value. The stack is always reserved in a LIFO order, the most recently reserved block is always the next block to be freed. This makes it really simple to keep track of the stack, freeing a block from the stack is nothing more than adjusting one pointer.

Variables allocated on the heap have their memory allocated at run time and accessing this memory is a bit slower, but the heap size is only limited by the size of virtual memory . Element of the heap have no dependencies with each other and can always be accessed randomly at any time. You can allocate a block at any time and free it at any time. This makes it much more complex to keep track of which parts of the heap are allocated or free at any given time.


Stack
int x=1
int y = 2
int x=1
int y = 2
Form1 frm = new Form1()
Heap
frm
Object

Sorted Heap
Unsorted Heap
Sorted & Stacked Piles

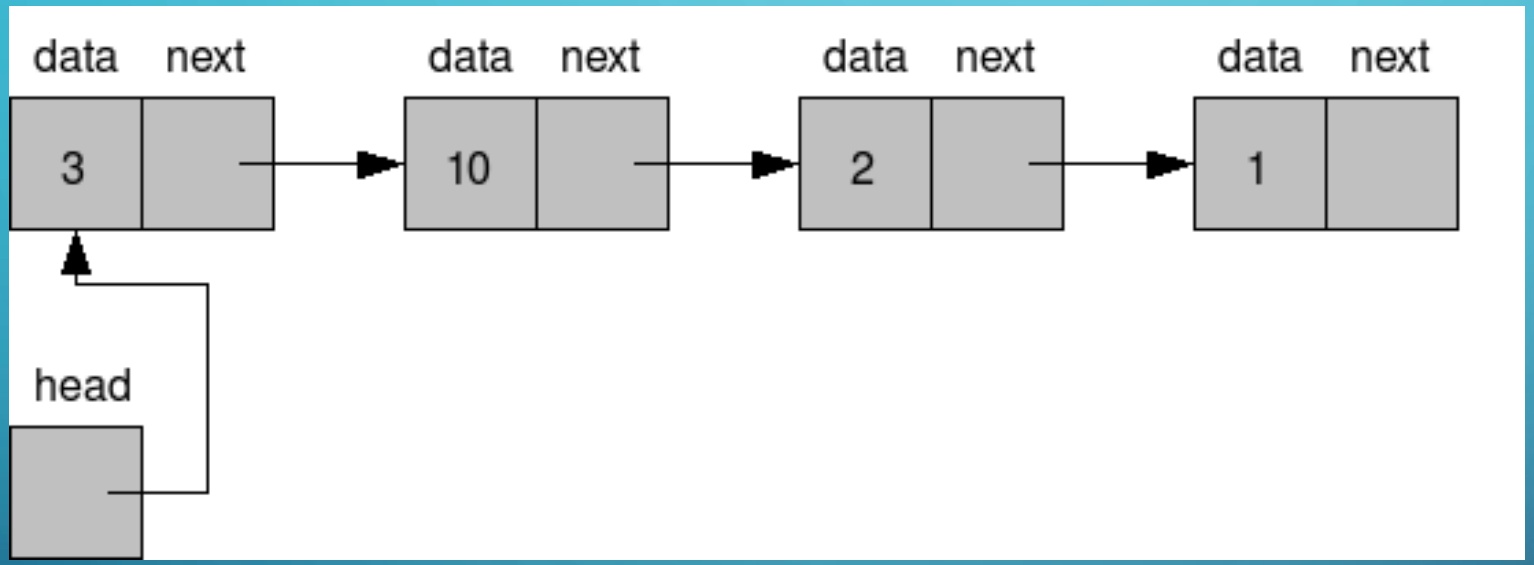
DYNAMIC MEMORY ALLOCATION

```
include <stdio.h>
#include <stdlib.h>

int main()
{
   char *str;

   /* Initial memory allocation */
   str = (char *) malloc(15);
   strcpy(str, "tutorialspoint");
   printf("String = %s,  Address = %u\n", str, str);
   free(str);

   return(0);
}
```

ADVANTAGES AND DISADVANTAGES OF ARRAY

- Arrays a limited by length
- But they are easy to create and use

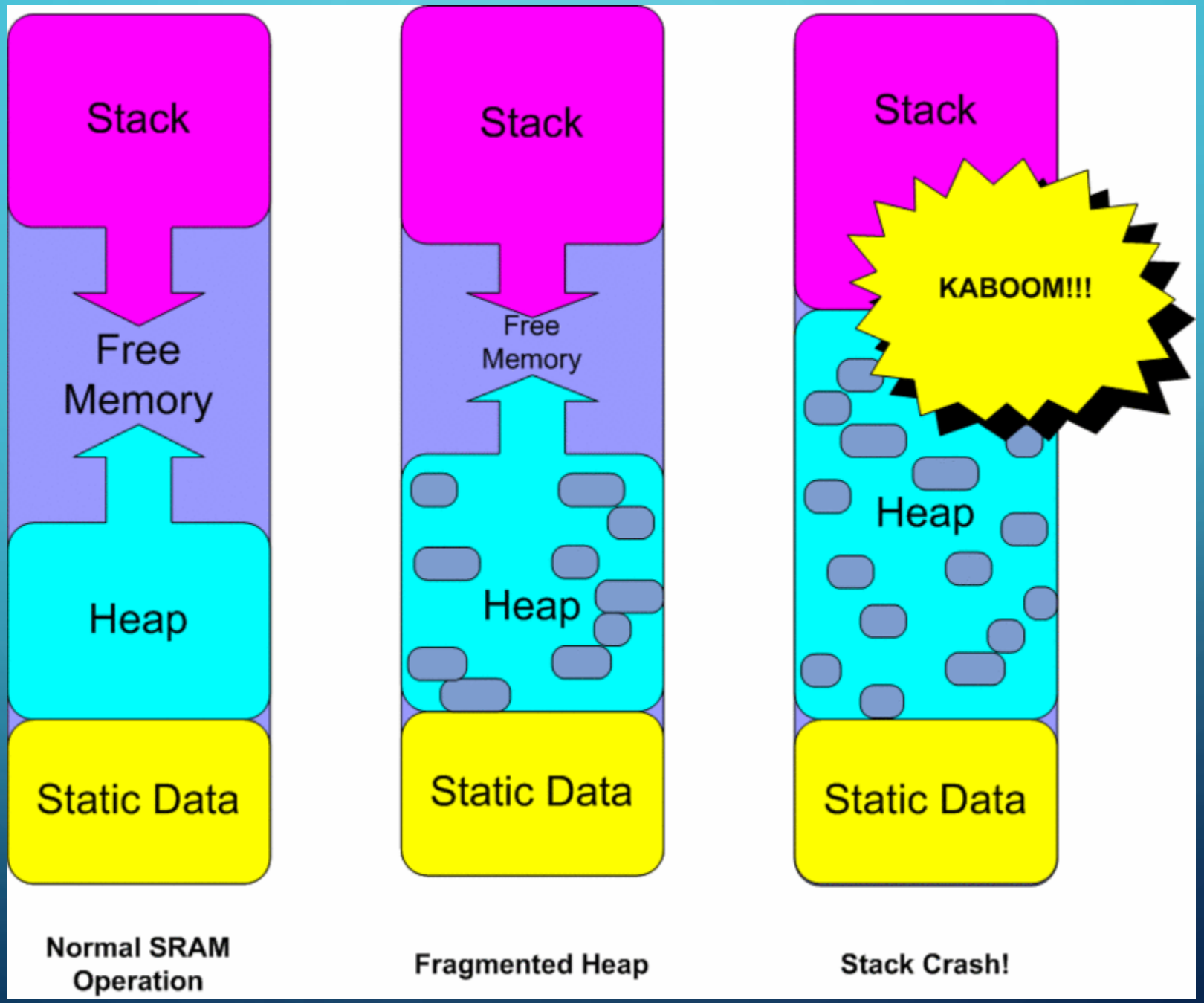
DYNAMIC DATA STRUCTURE AS A REPLACEMENT OF ARRAY




```
typedef struct node
{
int data
struct node *next
} NODE
```

HANDSON WITH LINKED LIST

RE-write all program which we had written using array with linked list


Stack
Free Memory
Heap
Static Data
Normal SRAM Operation
Stack
Free Memory
Heap
Static Data
Fragmented Heap
Stack
KABOOM!!!
Heap
Static Data
Stack Crash!